PERANCANGAN SISTEM PENDETEKSI KEPADATAN LALU LINTAS MENGGUNAKAN IMAGE PROCESSING DENGAN METODE BACKGROUND SUBTRACTION PADA SIKOMOLINTAS

Indra Hadi Setiadi[1], Yeffry Handoko P[2]
Jurusan Sistem Komputer, Fakultas Teknik dan Ilmu komputer, Universitas Komputer Indonesia
[1]indrahadisetiadi94@gmail.com, [2]Yeffry.handoko@gmail.com

ABSTRAK

Data jumlah kendaraan meningkat tiap tahunya, dampaknya dapat menyebabkan kepadatan lalu lintas yang berlebih dan apabila tidak ditindak lanjuti lebih jauh dapat menyebabkan kemacetan. Solusi yang dapat dilakukan yaitu dengan menambahkan kecerdasan pada pengendali pewaktuan lalu lintas dengan mendahulukan ruas jalan yang padat/macet. Namun tentu saja memerlukan data kepadatan untuk mengendalikanya. Tugas akhir ini bertujuan untuk membuat suatu sistem yang dapat mendeteksi kepadatan lalu lintas dan kondisi lalu lintas dengan menggunakan image processing. Metoda Background Subtraction digunakan sebagai pemisah objek dengan background, sedangkan kelemahan Background Subtraction yang memerlukan data tanpa objek pada latar belakang dibuat solusinya dengan menyamarkan objek bergerak. Sebagai contoh penerapan dari sistem ini dibuat juga simulasi pengendali lampu lalu lintas. Perancangan sistem ini merupakan bagian dari sistem yang lebih besar yaitu Sistem Kontrol dan Monitoring Lalu Lintas (SIKOMOLINTAS). Adapun hasil yang telah dicapai dari perancangan sistem ini diantaranya dapat menghasilkan data kepadatan dan status lalu lintas, mengendalikan pewaktuan simulasi lampu lalu lintas berdasarkan data kepadatan yang ada, serta dapat diintegrasikan dengan SIKOMOLINTAS

***Kata kunci**: lalu lintas, cuaca, suhu, kelembaban, hujan, kecerahan, SIKOMOLINTAS.*

1. PENDAHULUAN

1.1 Latar Belakang

Data dari badan pusat statistik [1] menyebutkan bahwa kenaikan jumlah kendaraan dari tahun 2010 sampai 2013 mengalami kenaikan sebesar 27.211.842 unit. Dengan adanya kenaikan jumlah kendaraan yang sangat besar tersebut tentunya menimbulkan masalah lalu lintas berupa kepadatan jalan raya yang berlebih. Apabila tidak ada antisipasi atau penanganan khusus, lebih jauhnya akan menimbulkan kemacetan Dampak negatif dari kemacetan diantaranya dapat menimbulkan keterlambatan bagi pengguna jalan, pemborosan energi, menigkatnya polusi udara pada suatu titik, meningkatkan stress bagi pengguna jalan, mengganggu kelancaran kendaraan darurat seperti ambulans dan pemadam kebakaran, menimbulkan penambahan biaya bahan bakar bagi pengendara (*increasing cost*). Solusi yang dapat dilakukan untuk menanggulangi kemacetan diantaranya dapat mengalihkan kendaraan ke jalur lain yang lebih lancar, serta menambahkan kecerdasan pada lampu lalu lintas. Namun solusi-solusi tersebut memerlukan data kepadatan lalu lintas untuk penerapanya.

Karena pentingnya data kepadatan lalu lintas untuk solusi-solusi diatas maka diperlukan suatu sistem yang dapat mendeteksi data kepadatan dan menghitung lalu lintas. Dengan menggunakan teknik pengolahan citra atau *Computer Vision* dapat menjadi solusi dengan berbagai keuntunganya.

Salah satu metode yang dapat digunakan dalam mendeteksi kepadatan lalu lintas yaitu metode *Background Subtraction*. Ide dari *Background Subtraction* yaitu melakukan pengurangan atau perbandingan *image* sekarang dengan *image reference* (*background*) [2]. Metode ini dipilih dengan menimbang keunikan dari lalu lintas yang dimiliki Indonesia, dimana semua jenis kendaraan bahkan bukan kendaraan dapat masuk ke jalan sehingga menimbulkan berbagai kondisi lalu lintas seperti macet, padat merayap, ramai lancar dan lancar. Dengan metode *Background Substraction* semua objek apapun yang berada pada area yang diinginkan dapat terdeteksi. Sedangkan kelemahan *Background Subtraction* yang memerlukan gambar jalan raya tanpa objek kendaraan apapun sebagai latar belakang

dibuat juga solusinya dengan menyamarkan objek bergerak yaitu algortima *Adaptive Background.* Selanjutnya untuk menjadikan data kepadatan lalu lintas, objek yang terdeteksi dihitung jumlah pikselnya dan dibandingkan dengan *frame* total. Hasil data kepadatan lalu lintas diatas dikirimkan melalui komunikasi *serial* untuk kepentingan pengaturan simulasi lampu lalu lintas dan sebagai informasi ke pengolah lain untuk di informasikan.

1.2 Tujuan

Maksud dari tugas akhir ini adalah membuat sistem pendeteksi lalu lintas menggunakan *image processing* dengan metoda *background subtraction* yang dapat memberikan nilai kepadatan lalu lintas dan status lalu lintas dengan komunikasi *serial*, serta dapat mengatur pewaktuan simulasi lampu lalu lintas.

Tujuan dari pembuatan tugas akhir ini adalah:

1. Menghasilkan data kepadatan lalu lintas dalam bentuk persen (%).
2. Menghasilkan data status lalu lintas
3. Memberikan informasi data kepadatan dan status lalu lintas berupa *serial*
4. Mengintegrasikan sistem dengan SIKOMOLINTAS

2. DASAR TEORI

2.1 Background Subtraction

Pendekatan yang umum digunakan identifikasi objek bergerak dari sekuen *video* dalam adalah teknik *Background Subtraction (BS)*. *BS* akan mengidentifikasi objek bergerak dari tiap *frame* yang berupa nilai pixel yang berbeda pada tiap *frame*. Ada beberapa tantangan dalam mengembangkan algoritma *BS*. Pertama algoritma harus tahan tehadap perubahan pencahayaan. Kedua, algoritma dapat menghindari deteksi objek non-stationer pada *background* seperti: daun bergerak, hujan, salju dan bayangan dari objek [6].

$$F(x, y) = It(x, y) - Bt(x, y) > T$$

Formula diatas merupakan notasi untuk mendeteksi objek pada *background subtraction* dimana t adalah *frame* masukan, Bt adalah model *background*, dan F adalah *foreground* atau objek bergerak yang terdeteksi. Selisih nilai *pixel* dari *frame* masukan dan model *background* akan dibandingkan dengan nilai *threshold*. Nilai *pixel* yang lebih dari nilai threshold akan dideteksi sebagai *foreground* dan sebaliknya dideteksi sebagai *background.*

2.2 OpenCV

Open Source Computer Vision (OpenCV) merupakan pustaka *open source* yang terdiri dari beberapa ratus algoritma *computer vision*. Opencv terdiri dari beberapa bagian modul seperti *core, imgproc, video, calib3d, feature2d, objdetect, highgui,* dan *gpu* [7]

2.3 Bahasa Pemograman Python

Python merupakan *general-purpose, high-level programming language* dengan kekuatan yang luar biasa dan sintaks yang sangat jelas dimana filosofi desainya menekankan pada kekuatan pembacaan kode sehingga mengurangi *cost* pemeliharaan program. Python satu-satunya bahasa pemograman yang menggunakan indentasi dalam pembuatan blok kodenya. Hal ini membuat kode python terlihat mirip terlepas dari siapapun yang menulisnya sehingga meningkatan kecepatan pembacaan kode [8].

Python mendukung beberapa paradigma pemrograman seperti berorientasi objek, pemrograman imperatif dan pemograman dengan cara fungsional (*functional programming style*). Python mempunyai *library* standar yang besar dan komperhensif. Fiturnya diantaranya yaitu sistem yang full dinamis dan mendeteksi siklus pengumpul sampah untuk manajemen memory yang otomatis, mirip seperti Scheme, Ruby, Pearl dan Tcl. Python sering digunakan sebagai bahasa scripting seperti bahasa dinamis lainnya, tetapi juga digunakan dalam berbagai konteks non-scripting. kode python dapat dikemas ke dalam program executable mandiri menggunakan alat pihak ketiga [9].

Python dirancang untuk menjadi sangat extensible yang tidak mewajibkan semua fungsi yang diinginkan dibangun ke dalam bahasa inti. Built-in modul baru dapat dengan mudah ditulis Baru built-in modul dapat dengan mudah ditulis dengan C, C++, atau Cython [10] . Python juga dapat digunakan sebagai bahasa ekstensi untuk modul yang sudah ada dan aplikasi yang membutuhkan sebuah *promgramable interface* [11].

2.4 Asyncronous Serial

Asyncronous serial adalah sebuah sistem komunikasi yang paling umum digunakan dalam berbagai macam sistem. Dengan menggunakan *Asyncronous serial*, kita dapat melakukan komunikasi antara satu *device* dengan *device* lainnya. Komunikasi ini terdiri dari dua komponen yaitu *transmiter* (TX) dan *receiver* (RX). Transmiter berfungsi untuk mengirimkan data kepada device target, sedangkan receiver berguna untuk menerima data dari device target.

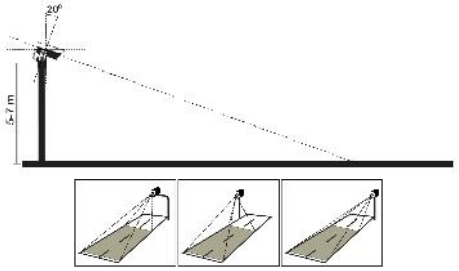
3. PERANCANGAN SISTEM

3.1 Perancangan Perangkat Keras

Perancangan perangkat keras pada sistem meliputi perancangan penempatan kamera dan perancangan *prototype* lalu lintas.

3.1.1 Perancangan Penempatan Kamera

Kamera menjadi penting karena bertindak sebagai sumber data oleh karena itu perancangan penempatan kamera diperlukan untuk mendapatkan data gambar yang baik. Mengikuti perancangan kamera pada publikasi internasional [16] kamera akan dipasang di jalan dengan ketinggian berkisar antara 5 hingga 7 meter dengan sudut kemiringan sekitar 20 derajat terhadap garis tegak. Posisi penempatan kamera terdapat 3 posisi yaitu tepat ditengah jalan, sebelah kiri dan sebelah kanan.

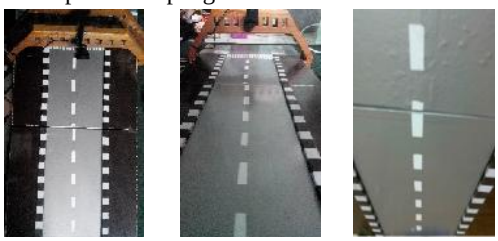


Gambar 3.1 Perancangan Penempatan Posisi Kamera Terhadap Jalan [16]

3.1.2 Perancangan Prototype Lalu lintas

Prototype lalu lintas dibuat untuk memudahkan pengujian dengan kamera secara langsung (bukan data dari *video*), sehingga membuktikan bahwa sistem sebenarnya sudah dapat diimplementasikan di jalan raya. Hanya saja keterbatasan pemroses yang dipakai belum portable dan penempatan kamera yang memerlukan izin dalam pemasanganya.

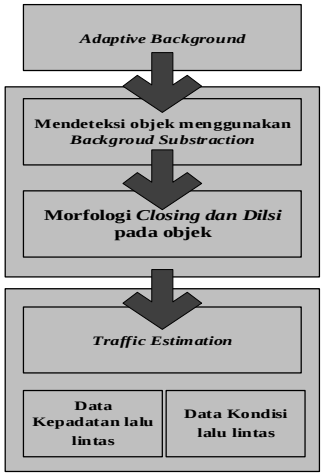
Kamera yang dipakai yaitu kamera USB yang resolusinya diperkecil menjadi 320 x 240 *pixel*. Hal ini bertujuan agar proses komputasi menjadi lebih singkat. Pemilihan ukuran ini berdasarkan pertimbangan dengan mengusahakan ukuran *video* sekecil mungkin, akan tetapi detail citra pada *video* masih dapat terbaca oleh komputer. Kamera akan bertindak sebagai sensor dengan menangkap gambar area pengamatan. Kemudian untuk memberikan variasi keadaan lalu lintas disimpan kendaraan mainan pada area pengamatan.



Gambar 3.2 Prototype *Lalu Lintas*

3.2 Perancangan Perangkat Lunak

Perancangan Perangkat Lunak secara garis besar dibagi menjadi 3 bagian utama seperti diperlihatkan pada , Bagian pertama yaitu *Image Acquisition* meliputi pengambilan *frame* yang dijadikan acuan menjadi *background* dengan menggunakan algoritma *Adaptive Background*, pada tahap ini gambar sudah diperkecil dan berskala keabuan. Bagian kedua yaitu deteksi dan perbaikan objek, meliputi pengurangan nilai *pixel frame* sekarang/*current frame* dengan *frame* acuan/*frame reference*, selanjutnya gambar dipotong sehingga area jalan saja yang dideteksi. Pada tahap ini juga gambar sudah dirubah bentuk menggunakan *morphology closing.* Pada bagian ketiga yaitu menghitung kepadatan/ *density count* yang diawali dengan mencari *pixel* berwarna putih (objek) dan menghitung jumlahnya Setelah jumlah *pixel* berwarna putih diketahui langkah selanjutnya yaitu melakukan perhitungan dengan formula persentasi *frame*, sehingga data kepadatan diketahui dalam satuan persen (%). Pada bagian terakhir data kepadatan dan status lalu lintas dikirimkan menggunakan komunikasi *serial* untuk keperluan pengendalian simulasi lampu lalu lintas.



Gambar 3.3 Diagram Blok Sistem

3.2.1 Perancangan Adaptive Background

Pada skripsi ini untuk mendapatkan model latar belakang yang baik sangat diperlukan. Hal tersebut menjadi penting karena metode yang dilakukan untuk mendeteksi objek yaitu menggunakan Background Subtraction. Sehingga gambar diperlukan yaitu gambar jalan raya dengan kondisi jalan yang kosong, atau dengan kata lain tidak ada kendaraan maupun objek apapun yang sering bergerak di jalan tersebut. Namun pada kenyataanya untuk mendapatkan kondisi ideal tersebut sangat sulit, karena pada kenyataanya kendaraan akan selalu muncul di jalan raya.

Solusi yang dibuat yaitu mendapatkan latar belakang dengan menggunakan algoritma *adaptive*

background. Dengan memanfaatkan fungsi AccumulatedWeight pada OpenCV [12] yang persamaan matematisnya sebagai berikut :

$$dst(x,y) \leftarrow (1-alpha)\cdot dst(x,y) + alpha \cdot src(x,y) \quad if \quad mask(x,y) \neq 0$$

Dimana :
src = Input image (1- atau 3-channel, 8-bit atau 32-bit floating point).
dst = Accumulator image with the same number of channels as input image, 32-bit or 64-bit floating-point.
mask = *Optional operation mask*.

Algoritma *adaptive background* akan aktif untuk melakukan *update background* setiap 5 menit sekali dengan kondisi kepadatan lalu lintas harus < 50%. Dan akan selesai ketika mencapai 100 frame.

3.2.2 Perancangan Pendeteksian Objek

Pendeteksian objek diperlukan untuk mendapatkan data/acuan yang akan dipakai untuk perhitungan kepadatan lalu lintas. Langkah yang dilakukan yaitu memisahkan objek dengan latar belakang/*background.* Dengan mengurangi nilai pixel kedua frame tersebut maka akan didapat perbedaan nilai, yang nantinya akan dijadikan sebagai pengambilan keputusan mengisi pixel tersebut. Dengan nilai 255 (putih) yang berarti objek atau 0 (hitam) yang berarti latar belakang/*background*.

Hasil dari proses diatas belum cukup, karena objek masih tipis dan akan mengurangi tingkat keberhasilan dari perhitungan yang akan dilakukan pada langkash berikutnya. Maka dilakukan morfologi *closing* untuk memperbaiki objek agar lebih tebal dan mengisi objek yang bolong. Hasil keluaran dari tahap ini berupa frame baru untuk keperluan tahap selanjutnya.

3.2.3 Traffic Estimation

Tahap terakhir dari sistem ini yaitu tahap *Traffic Estimation*, dimana keputusan untuk menentukan data kepadatan dan status lalu lintas dilakukan pada tahap ini. *Traffic Estimation* terdiri dari 3 langkah:

1. Pemisahkan objek dengan derau/*noise*
   Frame hasil tahap sebelumnya dibaca setiap pixelnya, lalu dibandingkan apakah pixel tersebut termasuk objek atau derau/*noise* dengan memanfaatkan *variable threshold*. Dalam pengecekanya nilai pixel untuk derau/*noise* akan kurang dari variabel *threshold* sedangkan untuk objek lebih dari *variable threshold.* Apabila pixel tersebut objek maka akan dihitung, sehingga pada tahap akhirnya akan memunculkan jumlah pixel yang dianggap sebagai objek dari frame tersebut.

   F (x,y) = 255 ; untuk P(x,y) > threshold, x<width, y<height
   JPP = JPP + 1
   F (x,y) = 0 ; untuk P(x,y) < threshold, x<width, y<height

   Dimana :
   JPP = Jumlah Pixel Putih
   P(x,y) = Koordinat Pixel pada frame yang akan dibandingkan
   F(x,y) = Koordinat Pixel pada frame setelah dibandingkan
   Width = lebar gambar
   Height = Tinggi gambar
2. Pengecekan untuk zonasi pixel
   Pada sistem ini penambahan zonasi digunakan untuk membantu menentukan status macet pada lalu lintas. Idenya yaitu dengan mengecek kepadatan zonasi depan dan zonasi belakang, karena secara visual manusia bisa dikatakan bahwa jika bagian depan dan belakang jalan padat dan kepadatan total jalan berlebih setelah 5 menit, maka secara visual manusia dapat dikatakan macet.
   Setelah melakukan pengecekan pada langkah 1, pixel yang dianggap objek akan dibandingkan apakah masuk pada zonasi atau tidak. Zonasi jalan sendiri dibagi menjadi dua yaitu zona depan dan zona belakang.

   JPP_F/B= JPP_F/B + 1 ; untuk P(x,y) > threshold, x<width_F/B, y<height_F/B

   Dimana :
   JPP_F/B = Jumlah Pixel Putih zona depan/belakang
   P(x,y) = Koordinat Pixel
   Width = Lebar zonasi depan/belakang
   Height = Tinggi zonasi depan/belakang
3. Penentuan Data Kepadatan Lalu Lintas
   Data kepadatan lalu lintas pada sistem ini ditentukan dengan melakukan perbandingan antara jumlah pixel putih dengan luas pixel frame, lalu hasil perbandingan tersebut dirubah dalam bentuk persen (%). Secara matematis dapat ditulis sebagai berikut :

$$K \quad (\%) = \frac{Ju \quad h\,t\iota \quad p \quad p \quad h}{L \quad p \quad f} \text{ x } 100\%$$

$$K \quad (\%) = \frac{Ju \quad ht \quad p \quad p \quad hz \quad d}{L \quad p \quad f \quad z \quad d} \times 100\%$$

$$K \quad (\%) = \frac{Ju \quad ht \quad p \quad pu \quad hz \quad b}{L \quad p \quad f \quad z \quad b} \times 100\%$$

Dimana :

KPLL = Kepadatan lalu lintas keseluruhan yang diamati sistem

KPLF = Kepadatan lalu lintas zona depan yang diamati sistem

KPLB = Kepadatan lalu lintas zona belakang yang diamati sistem

4. Penentuan Status Kepadatan Lalu Lintas

Setelah melakukan observasi dilapangan dan melakukan wawancara dengan Polisi lalu-lintas, penentuan statu lalu lintas diambil dari pengamatan selama waktu tertentu. Sehingga pada sistem ini data status kepadatan lalu lintas ditentukan dengan melakukan rata-rata selama 5 menit dari data kepadatan lalu lintas. Kondisi lalu lintas ditentukan berdasarkan parameter seperti pada tabel 1 perlihatkan.

Status Kepadatan Lalu lintas	Jumlah rata rata kepadatan total (X)	Jumlah rata-rata kepadatan depan (Y)	Jumlah rata-rata kepadatan belakang (Z)
Macet	X > 70 %	Y > 70%	Z > 70 %
Padat merayap	55 %< X <= 70 %	55 % < Y< =70 %	55 % < Z <= 70 %
Ramai lancar	50 % < X <= 55 %	30 % < Y <= 55 %	30 % < Y <= 55 %
Lancar	X <=50%	X <= 30%	Z <=30

4. HASIL PENGUJIAN DAN ANALISA

Pada bab ini akan dijelaskan hasil dan analisis pada sistem yang dibuat. Setiap sub-sistem di uji apakah sudah sesuai dengan fungsinya atau belum, serta pengujian keandalan sistem dengan menambahkan dan mengurangi kontras pada *frame*/gambar. Data yang akan dipakai untuk pengujian berupa data video hasil rekam kamera digital yang sudah dikecilkan ukurannya menjadi 320 x 240 *pixel*. Proses perekaman video diambil diatas jembatan penyebrangan Pasteur Bandung dengan kondisi lalu lintas sebenarnya.

4.1 Adaptive Background

Mengingat metoda Background Subtraction memerlukan background yang baik atau tidak ada objek lain pada area pengamatan gambar. Maka dibuat adaptive background untuk merata-ratakan bobot tiap gambar, sehingga mendapatkan background jalan raya dengan objek bergerak sebelumnya ada menjadi samar. Pada tahap ini sistem dapat mengerjakan proses tersebut dengan hasil yang ditunjukan oleh gambar dibawah.

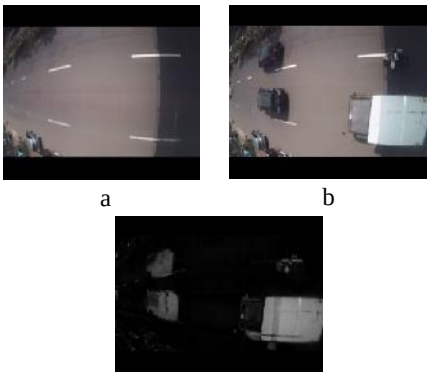


A b

Gambar 4.1 Hasil Pengujian Adaptive Background

4.2 Deteksi Objek Menggunakan Background Subtraction

Pendeteksian objek diperlukan untuk mendapatkan data/acuan yang akan dipakai untuk perhitungan kepadatan lalu lintas. Langkah yang dilakukan yaitu memisahkan objek dengan latar belakang/*background.* Pada tahap ini sistem dapat mengerjakan proses tersebut dengan hasil yang ditunjukan oleh gambar 5.

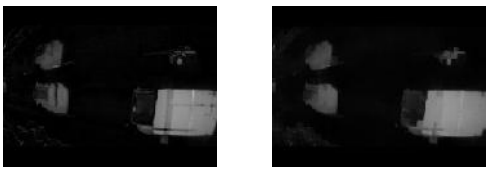


a b

Gambar 4.2 Hasil Pengujian Background Subtraction

4.3 Perbaikan Objek dengan Morfologi

Hasil dari proses diatas belum cukup, karena objek masih tipis dan akan mengurangi tingkat keberhasilan dari perhitungan yang akan dilakukan pada langkash berikutnya. Maka dilakukan morfologi *closing* untuk memperbaiki objek agar lebih tebal dan mengisi objek yang bolong. Pada tahap ini sistem dapat mengerjakan proses tersebut dengan hasil yang ditunjukan oleh



a b

Gambar 4.3 Hasil Pengujian Morfologi

4.4 Penentuan Data Kepadatan Lalu Lintas

Pada sistem ini data status kepadatan lalu lintas ditentukan dengan melakukan rata-rata selama 5 menit dari data kepadatan lalu lintas. Adapun hasil pengujianya seperti tabel dan gambar dibawah .

Tabel 4.1 Hasil pengujian kepadatan lalu lintas

Frame ke	KPLL_Total (%)	KPLLF_Total (%)	KPLLB_ Total (%)	Status
800	36.3	43.5	32	Lancar
850	36.3	43.5	32	Lancar
900	36.3	43.5	32	Lancar
950	36.3	43.5	32	Lancar
1000	47.5	53.4	43	Ramai Lancar
1050	47.5	53.4	43	Ramai Lancar
1100	47.5	53.4	43	Ramai Lancar
1150	47.5	53.4	43	Ramai Lancar
1200	40.8	47.0	35.8	Lancar
1250	40.8	47.0	35.8	Lancar



A b

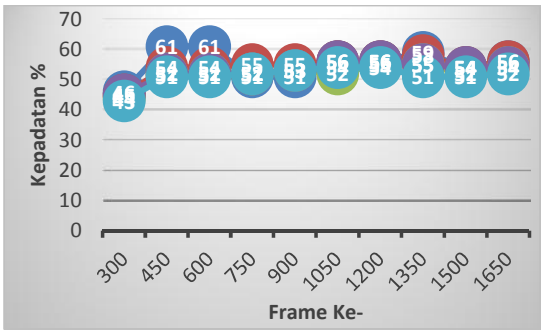
Gambar 4.4 Contoh sampel frame yang diuji (a) status lancar dan (b) status ramai lancar

Pada pengujian ini sistem secara keseluruhan akan di uji keandalanya dengan cara menambah dan mengurangi kontras gambar/*frame*. Perlu diketahui sebelumnya bahwa penambahan kontras menggunakan fungsi [11]:

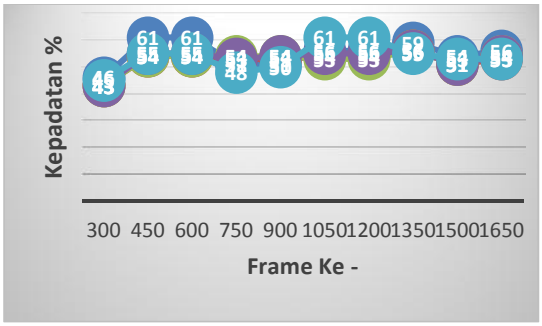
cv2.multiply(f,np.array([alpha]))

Dimana mode normal (tanpa penambahan cahaya) nilai *alpha* adalah 1. Untuk keperluan pengujian, *variable alpha* diubah menjadi naik (menambah kontras) dari mulai 1.2 sampai 2, dan turun (mengurangi kontras) dari mulai 0.8 sampai 0.2. Untuk mendapatkan hasil yang valid (mendekati dengan mode normal/tanpa penambahan cahaya) maka sistem dibuat secara otomatis menyesuaikan variabel threshold dengan kecerahan gambar. Pada tahap ini sistem dapat mengerjakan proses tersebut dengan hasil yang ditunjukan oleh tabel 4.4.1, 4.4.2, 4.4.3, 4.4.4, 4.4.5, 4.4.6, 4.4.7 dan 4.4.8.

Pada pengujian ini dapat dianalisis bahwa penambahan kontras dan penurunan kontras berbanding lurus dengan nilai *threshold*. Dimana ketika *alpha* 1.8 (pengujian kecerahan paling maksimum) *threshold* sebesar 70, sedangkan ketika *alpha* 0.2 (pengujian kecerahan paling minimum) *threshold* sebesar 18.



Gambar 4.5 Grafik Data Kepadatan Jika Kontras Dinaikan



Gambar 4.6 Grafik Data Kepadatan Jika Kontras Ditrunkan

4.4.1 Pengujian Komunikasi Serial

Setelah data kepadatan lalu lintas didapat, berikutnya sistem harus dapat mengirimkan data kepadatan lalu lintas dan menerima data parameter cahaya dari pengolah lain (pada pengujian ini paramere cahaya masih *dummy*). Data kepadatan lalu lintas digunakan untuk mengendalikan simulasi lampu lalu lintas dan memenuhi kepentingan pengolah lain (sdr. Angga Anggara) untuk informasi kepadatan lalu lintas. Pada pengujian ini juga dilakukan dengan mnggunakan *prototype* lampu lalu lintas dengan kendaraan mainan sebagai objeknya dengan tujuan variasi dari kemacetan dapat diubah-ubah dengan mudah.

Pada tahap ini sistem berhasil melakukan komunikasi *serial* dengan hasil sebagai berikut:

Pengujian ke-	Data kepadatan yang dikirm	Status Kepadatan Yang dikirim	Data cahaya yang diterima	Status
1	21	Lancar	25	Berhasil
2	38	Lancar	65	Berhasil
3	9	Lancar	40	Berhasil
4	9	Lancar	47	Berhasil
5	0	Lancar	47	Berhasil
6	55	Ramai Lancar	60	Berhasil
7	69	Padat Merayap	20	Berhasil
8	80	Macet	54	Berhasil
9	75	Macet	40	Berhasil
10	90	Macet	22	Berhasil

Pengujian ke-	Data kepadatan yang dikirm	Status Kepadatan Yang dikirim	Data cahaya yang diterima	Status
Jumlah berhasil				10

5. KESIMPULAN

1. Sistem dapat mengambil data gambar lalu lintas, mengolah, dan menghasilkan data kepadatan serta status lalu lintas. Data tersebut selanjutnya dapat dikirimkan melalui komunikasi *serial*, untuk diterapkan pada simulasi pengendali lampu lalu lintas dengan hasil pewaktuan sudah fleksibel berdasarkan data kepadatan yang ada.
2. Metoda *Background Subtraction* sudah dapat diaplikasikan pada penerapan pendeteksi kepadatan lalu lintas dengan keandalan sistem masih andal apabila tingkat kecerahan dinaikan (pada pengujian data *video*) sampai dengan *alpha* 1.8 dengan rata-rata *error* 3.38% dan kesalahan status kemacetan terjadi 1 kali. Sedangkan apabila tingkat kecerahan diturunkan (pada pengujian data *video*) sampai dengan *alpha* 0.2 dengan rata-rata *error* 3.4% dan kesalahan status kemacetan tidak ada.
3. Berdasarkan hasil pengujian integrasi, sistem yang dibuat sudah dapat diintegrasikan (dengan sistem yang dibuat oleh sdr. Angga Anggara). Hasil integrasi sistem tersebut dapat menghasilkan sebuah produk yaitu SIKOMOLINTAS.

6. DAFTAR PUSTAKA

[1] "Badan Pusat Statistik : Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis tahun 1987-2013," Badan Pusat Statistik, 5 Desember 2014. [Online]. Available: http://bps.go.id/linkTabelStatis/view/id/1413. [Accessed 16 Juni 2016].

[2] Federal Highway administration, "Traffic Congestion and Reability," CAMBRIDGE SYSTEMATICS, Cambridge, 2005.

[3] N. N. J and A. B. K, "Role of Computer Vision in Automatic Inspection," *International Journal of Computer Applications (0975 – 8887)*, vol. 123, no. 13, pp. 28-31, 2015.

[4] K. Kyungnam, T. H. Chalidabhongse, D. Harwood and L. Davis, "Real-time foreground–background segmentation," 31 March 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077201405000057. [Accessed 16 Juni 2016].

[5] A. Kadir and A. Susanto, Teori dan Aplikasi Pengolahan Citra, Yogyakarta: Andi, 2013.

[6] R. Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung: Informatika, 2004.

[7] G. Bradski and A. Kaehler, Learning OpenCV, California: O'Reilly Media, Inc, 2008.

[8] R. C. Gonzales, R. E. Woods and S. L. Eddins, Digital Image Processing, New Jersey: Prentice Hall, 2008.

[9] M. J. Burge and W. Burger, Digital Image Processing An Algortimic Introdicing using Java, New York: Springer Science + Business Media,LLC., 2008.

[10] N. Wahyudi, V. Suhartono and R. A. Pramunendar, "BACKGROUND SUBTRACTION BERBASIS SELF ORGANIZING," *SYSTEMIC*, vol. 1 No.2, pp. 42-51, 2015.

[11] o. d. team, The OpenCV Reference Manual (Release 2.4.9), 2014.

[12] K. J. Millman and M. Aivazis, "Python for scientists and engineers," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 9-12, 2011.

[13] M. G. Rashed and R. Ahsan, "Python in Computational Science: Applications and Possibilites," *International Journal of Computer Applications (0975 – 8887)*, vol. 46 No.20, pp. 26-30, 2012.

[14] S. B. Benhel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn and K. Smith, "Cython: The best of both worlds," *Computing in Science & Engineering*, vol. 13, pp. 31-39, 2011.

[15] F. Perez, B. E. Granger and J. D. Hunter, "Python: an ecosystem for scientific computing," *Computing in Science & Engineering*, vol. 2, pp. 13-21, 2011.

[16] T. N. Nizar, "Multi-object tracking and detection system based on feature detection of the intelligent transportation system," in *System Engineering and Technology (ICSET), 2014 IEEE 4th International Conference on (Volume:4)*, Bandung, 2014.

[17] N. Berlian, E. Limberdo and J. , "PERANCANGAN MEDIA PENYAMPAIAN INFORMASI OTOMATIS," in *Skripsi S1*, Jakarta, 2010.

[18] Directorate General Bina Marga, "INDONESIAN HIGHWAY CAPACITY," Jakarta, 1997.

[19] B. S. Koloway, "KINERJA RUAS JALAN PERKOTAAN JALAN PROF DR. SATRIO,," *Jurnal Perencanaan Wilayah dan Kota*, vol. 20, no. 3, pp. 215-230, 2009.

[20] opencv dev team, "Motion Analysis and Object Tracking," [Online]. Available: http://docs.opencv.org/2.4/modules/imgproc/doc/motion_analysis_and_object_tracking.html. [Accessed 16 Juni 2016].

[21] M. T. T. Q. Naeem Abbas, "Real Time Traffic Density Count using Image Processing," *International Journal of Computer Applications*, vol. 83, pp. 16-19, 2013.

[22] S. S. Harsa and C. H. Sandeep, "Real Time Traffic Density And Vehicle Count Using Image," *International Journal of Research in Computer and*, vol. 4, no. 8, pp. 594-598, 2015.